



# From Modular to Distributed Architectures: Centaur and Smeci Experiments

Anne-Marie Dery, Wided Lejouad

## ► To cite this version:

Anne-Marie Dery, Wided Lejouad. From Modular to Distributed Architectures: Centaur and Smeci Experiments. [Research Report] RR-2398, INRIA. 1994. inria-00074277

**HAL Id: inria-00074277**

**<https://inria.hal.science/inria-00074277>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***From Modular to Distributed Architectures:  
Centaur and Smeci Experiments***

Anne-Marie Dery, Wided Lejouad

**N° 2398**

Novembre 1994

PROGRAMME 2

Calcul symbolique,  
programmation  
et génie logiciel

 ***rapport  
de recherche***

**1994**





# From Modular to Distributed Architectures: Centaur and Smeci Experiments

Anne-Marie Dery\*, Wided Lejouad\*\*

Programme 2 — Calcul symbolique, programmation et génie logiciel  
Projet Secoia

Rapport de recherche n° 2398 — Novembre 1994 — 25 pages

**Abstract:** In this report we point out the reasons that lead us to distribute modular applications, why we think that modularity is a critical step – but not a final one – in the design of distributed applications, and how we distribute. We illustrate our ideas with two experiments developed in the CROAP and SECOIA projects at INRIA\*\*\*. They concern respectively CENTAUR and SMECI tools.

**Key-words:** Programming environments, expert systems, distributed systems, communication protocols, client-server architecture.

*(Résumé : tsvp)*

\*Anne-Marie.Pinna@essi.fr

\*\*Wided.Lejouad@sophia.inria.fr

\*\*\*Institut National de Recherche en Informatique et Automatique.

# **Le passage d'une architecture modulaire à une architecture distribuée : exemples des systèmes Centaur et Smeci**

**Résumé :** Dans ce rapport, nous présentons les raisons qui nous ont amené à répartir des applications modulaires existantes, pourquoi nous pensons que la modularité est une étape essentielle mais non suffisante dans la conception d'applications distribuées et nous décrivons la démarche que nous avons suivie pour réaliser cette distribution. Nous illustrons ces points par nos deux expériences qui concernent les outils CENTAUR et SMECI développés respectivement dans les projets CROAP et SECOIA.

**Mots-clé :** Environnements de programmation, systèmes experts, systèmes distribués, protocoles de communication, architecture client-serveur.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Description of used tools</b>	<b>4</b>
2.1	Smeci Experiment . . . . .	4
2.2	Centaur Experiment . . . . .	5
2.3	Common objectives . . . . .	6
<b>3</b>	<b>From modularity to distribution</b>	<b>7</b>
3.1	Modularity . . . . .	7
3.2	Distribution Goals . . . . .	9
3.2.1	Autonomy . . . . .	9
3.2.2	Parallelism . . . . .	9
3.2.3	Reuse and heterogeneity . . . . .	10
3.2.4	Data sharing and Collaboration . . . . .	11
3.2.5	Maintenance . . . . .	11
3.3	Problems faced with distribution . . . . .	12
<b>4</b>	<b>Distributed Application Design</b>	<b>14</b>
4.1	Modularity step . . . . .	14
4.2	Communication . . . . .	15
4.2.1	Communication Types . . . . .	15
4.2.2	Communication protocols . . . . .	17
4.3	Process synchronization . . . . .	19
<b>5</b>	<b>Conclusion</b>	<b>20</b>

## 1 Introduction

Distributed systems began to appear in the 1970s, following the availability of computer networks. Since, the key computer technology has been information gathering, processing, and distribution [37] [10]. The idea of distribution is relevant to many areas of computing, indeed all of them. We can find distributed database systems, distributed artificial intelligence systems, distributed software engineering tools, etc. Perhaps the most well-known distributed application in the computer world is the X Window system [6].

In this report, we discuss the common concepts as well as the most striking differences between modular and distributed systems. We describe the advantages of distribution, and some reasons for choosing this approach. We also offer some guidelines for the design of such distributed applications. We illustrate each of our points with concrete examples from our design and implementation experience with the distribution of two applications, a programming environment generated by the CENTAUR system [4] [18] and the SMECI expert system shell [19].

## 2 Description of used tools

In addition to SMECI and CENTAUR, which will be described in detail in paragraphs 2.1 and 2.2, we work on SUN workstations under the UNIX operating system and an X Window environment. So we are concerned with neither parallel languages nor parallel machines.

### 2.1 Smeci Experiment

SMECI is an expert system shell used for design and planning, which is considered as a multiformalism environment for developing knowledge based applications. It groups several concepts, namely, frames, inference rules, tasks and states. It has been chosen as an example for distributing an expert system shell.

The frame language is needed to structure knowledge, the rule language infers new facts in the knowledge base. The task-oriented programming allows the decomposition of one problem into several sub-problems, and consequently